3

Exporting symbolic objects to databases

Donato Malerba, Floriana Esposito and Annalisa Appice

3.1 The method

SO2DB is a SODAS module that exports a set of symbolic objects (SOs) to a relational database and determines the individuals in a population Ω that are part of the extent of these SOs. Detailed data (micro-data) on individuals in Ω are stored in a relational database. Therefore, SO2DB is the counterpart of DB2SO which imports the descriptions of SOs by generalizing micro-data stored in a relational database.

Recall from Bock and Diday (2000) and Diday and Esposito (2003) that an SO is defined by a description d, a relation R which compares d to the description d_w of an individual, and a mapping a called the 'membership function'. Hence, the extent of an SO s, denoted by Ext(s) in the Boolean case (i.e., $[y(w)\ R\ d] \in \{\text{true, false}\}$), is defined as the set of all individuals w from a population Ω with $a(w) = \text{true}$. It is identical to the extension of a, denoted by Extension(a). Hence, we have $\text{Ext}(s) = \text{Extension}(a) = \{w \in \Omega | a(w) = \text{true}\}$. Conversely, in the probabilistic case (i.e., $[y(w)\ R\ d] \in [0,1]$), given a threshold α, the extent of an SO s is defined by $\text{Ext}_\alpha(s) = \text{Extension}_\alpha(a) = \{w \in \Omega | a(w) \geq \alpha\}$.

The extent of an SO that is computed on a population Ω can be used either to manage the information that is lost during the import process from Ω or to study the evolution of the retained concept on the same population Ω at a different time. Alternatively, comparing the extents of the same SO computed on several populations (e.g., populations associated with different countries) can be used to investigate the behaviour of some phenomenon in different regions. For instance, let us suppose that the following description associated with an SO s,

$$[gender = \text{F}] \wedge [field = \text{factory}] \wedge [salary = [1.5, 2.1]] \\ \wedge [weekly\_working\_hours = [40, 42]],$$

Symbolic Data Analysis and the SODAS Software Edited by E. Diday and M. Noirhomme-Fraiture

is obtained by generalizing data collected by the Finnish National Statistics Institute on a sample of the Finnish female population working in a factory. In this case, the data analyst may determine the extent of *s* computed on the populations of other European countries and perform a comparative analysis of these countries on the basis of working conditions of women in factories. This operation, supported by SO2DB and known as *propagation on a database*, is generally useful for discovering new individuals stored in databases of individuals different from those used to generate an SO but with similar characteristics.

The SO2DB module is run by choosing Export . . . from under Sodas file in the SODAS menu bar. This module inputs a set of SOs stored in an ASSO file and matches each SO against the individuals in a population Ω (micro-data) stored in a relational database. The correct association between the names of symbolic variables and the attributes of the relational database is user-defined. The method returns a new database table describing the matching individuals in Ω. Attributes of the new relational table are denoted with the names of the input symbolic variables.

Input, matching procedure and output are detailed in the next sections.

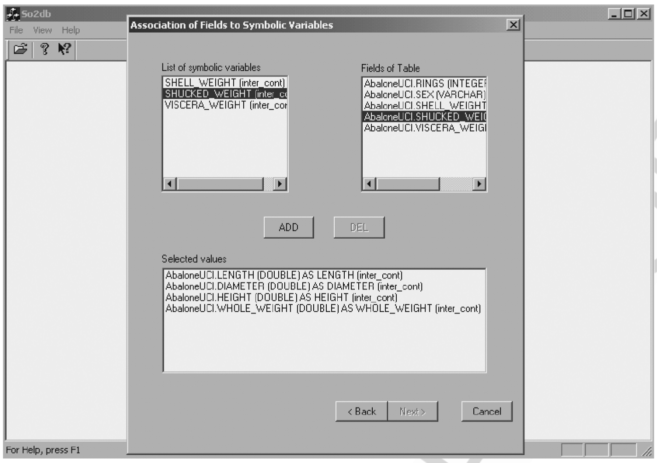
3.2 Input data of SO2DB

The input ASSO file contains a symbolic data table, whose columns correspond to symbolic variables, while rows represent symbolic descriptions d. An SO $s = (d, R, a)$ corresponding to a description d models the underlying concept and it is assumed to be in the form of assertion. Individuals of a population Ω are stored in a single table of a relational database. The current release of SO2DB has been tested on an MS Access database which is accessed through an ODBC driver.

The association between symbolic variables reported in the input ASSO file and columns of the database table is interactively established by the user by means of a graphical user interface (see Figure 3.1). The association may actually involve only a subset of symbolic variables. The user is allowed to perform a selection in order to link each symbolic variable to one table column. In any case, both subsets must have the same cardinality and the user-defined association must be a bijective function. The association of a symbolic variable with a table column is admissible in the following cases:

Type of symbolic variable	*Type of table column*
categorical single-valued	string
categorical multi-valued	string
quantitative single-valued	number (integer or real)
interval	number (integer or real)
modal	string

If all selected symbolic variables are either categorical single/multi-valued or quantitative single-valued or interval then input SOs are handled as boolean symbolic objects (BSOs) (see Chapter 8). If all selected symbolic variables are modal then input SOs are handled as probabilistic symbolic objects (PSOs). As explained in the next section, matching functions



List of symbolic variables

SHELL_WEIGHT (inter_cont)
SHUCKED_WEIGHT (inter_co
VISCERA_WEIGHT (inter_cor

Fields of Table

AbaloneUCI.RINGS (INTEGEI
AbaloneUCI.SEX (VARCHAR)
AbaloneUCI.SHELL_WEIGHT
AbaloneUCI.SHUCKED_WEI(
AbaloneUCI.VISCERA_WEIGI

Selected values

AbaloneUCI.LENGTH (DOUBLE) AS LENGTH (inter_cont)
AbaloneUCI.DIAMETER (DOUBLE) AS DIAMETER (inter_cont)
AbaloneUCI.HEIGHT (DOUBLE) AS HEIGHT (inter_cont)
AbaloneUCI.WHOLE_WEIGHT (DOUBLE) AS WHOLE_WEIGHT (inter_cont)

Figure 3.1 An example of an user-interactive association between the symbolic variables reported in an ASSO file and columns of a database in SO2DB.

implemented in SO2DB are defined for either BSOs or PSOs. In the general case of SOs described by both set-valued and modal variables, a combination of a matching function for BSOs with a matching function for PSOs is used to compute the extent of input SOs.

Finally, SO2DB allows users to select a subset of rows of the symbolic data table to match against individuals of the population Ω. In this case, the extent will be computed only for a subset of input SOs.

3.3 Retrieving the individuals

The exportation of an SO s to a population Ω aims to retrieve the individuals (micro-data) of Ω which are 'instances' of the class description underlying s (i.e., the extent of s computed on Ω). Let us consider:

- an SO s whose description is of the form

$$s : [Y_1 \in v_1] \wedge [Y_2 \in v_2] \wedge \ldots \wedge [Y_m \in v_m],$$

  where each $Y_i (i = 1, \ldots, m)$ is a symbolic variable,
- a population Ω of individuals described by m single-valued (continuous and discrete) attributes Y_i' such that there is a correct association between the name of the symbolic variables Y_i and the attribute Y_i'.

The extent of s is computed by transforming each individual $I \in \Omega$ in an SO s_I that is underlying the description of I and resorting to a matching judgement to establish whether the individual I described by s_I can be considered as an instance of s.

The matching between SOs is defined in Esposito *et al.* (2000) as a directional comparison involving a referent and a subject. The referent is an SO representing a class description, while the subject is an SO that typically corresponds to the description of an individual.

In SODAS, two kinds of matching are available, namely, *canonical matching* and *flexible matching*. The former checks for an exact match, while the latter computes the degree of matching that indicates the probability of precisely matching the referent against the subject, provided that some change is possibly made in the description of the referent.

Canonical matching is defined on the space S of BSOs as follows:

$$CanonicalMatch: S \times S \rightarrow \{0, 1\}.$$

This assigns the value 1 or 0 as result of the matching comparison of a referent r against a subject s. The value is 1 (0) when the individual described by the subject is (not) an instance of the concept defined by the referent. More precisely, let us consider the following pair:

$$r: [Y_1 \in R_1] \wedge [Y_2 \in R_2] \wedge \ldots \wedge [Y_p \in R_p],$$
$$s: [Y_1 \in S_1] \wedge [Y_2 \in S_2] \wedge \ldots \wedge [Y_p \in S_p].$$

Then

$$CanonicalMatch(r, s) = \begin{cases} 1, & \text{if } S_j \subseteq R_j \ \forall j = 1, \ldots, p, \\ 0, & \text{otherwise.} \end{cases}$$

Similarly, flexible matching is defined on S by

$$FlexMatch: S \times S \rightarrow [0, 1],$$

such that

$$FlexMatch(r, s) = \max_{s' \in S(r)} P(s|s'),$$

where $S(r) = \{s' \in S | CanonicalMatch\ (r, s') = 1\}$ and P represents the probability (likelihood) that the observed subject is s when the true subject is s' . The reader may refer to Esposito *et al.* (2000) for more details on both canonical matching and flexible matching.

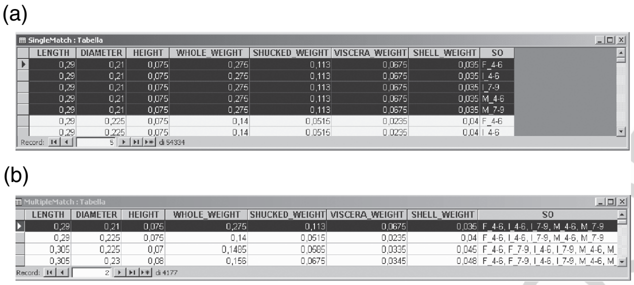
Finally, the definition of flexible matching can be extended to the space of PSOs as described in Chapter 8.

Notice that, in the case of canonical matching comparison, exporting s to Ω retrieves the individuals $I \in \Omega$ for which $CanonicalMatch(s, s_I) = 1$, while in the case of flexible matching comparison, given a user-defined threshold $fm\text{-}Threshold \in [0, 1]$, exporting s to Ω retrieves the individuals $I \in \Omega$ such that $FlexMatch(s, s_I) \geq fm\text{-}Threshold$.

3.4 Output of SO2DB

SO2DB outputs a new database table that describes the matching individuals in Ω. This table includes both attributes denoted with the names of the input symbolic variables and an additional attribute denoted with 'SO'. Rows of this table describe the individuals in Ω, which have at least one SO matching them.

The matching comparison is performed using either canonical or flexible matching as specified by the user. In particular, for each individual in Ω, the result of the matching comparison is stored in the 'SO' attribute as either one record for each single matching



(a)

SingleMatch : Tabella

LENGTH	DIAMETER	HEIGHT	WHOLE_WEIGHT	SHUCKED_WEIGHT	VISCERA_WEIGHT	SHELL_WEIGHT	SO
0,29	0,21	0,075	0,275	0,113	0,0675	0,035	F_4-6
0,29	0,21	0,075	0,275	0,113	0,0675	0,035	I_4-6
0,29	0,21	0,075	0,275	0,113	0,0675	0,035	I_7-9
0,29	0,21	0,075	0,275	0,113	0,0675	0,035	M_4-6
0,29	0,21	0,075	0,275	0,113	0,0675	0,035	M_7-9
0,29	0,225	0,075	0,14	0,0515	0,0235	0,04	F_4-6
0,29	0,225	0,075	0,14	0,0515	0,0235	0,04	I_4-6

Record: 5 di 54334

(b)

MultipleMatch : Tabella

LENGTH	DIAMETER	HEIGHT	WHOLE_WEIGHT	SHUCKED_WEIGHT	VISCERA_WEIGHT	SHELL_WEIGHT	SO
0,29	0,21	0,075	0,275	0,113	0,0675	0,035	F_4-6, I_4-6, I_7-9, M_4-6, M_7-9
0,29	0,225	0,075	0,14	0,0515	0,0235	0,04	F_4-6, I_4-6, I_7-9, M_4-6, M_7-9
0,305	0,225	0,07	0,1485	0,0585	0,0335	0,045	F_4-6, F_7-9, I_4-6, I_7-9, M_4-6, M_
0,305	0,23	0,08	0,156	0,0675	0,0345	0,048	F_4-6, F_7-9, I_4-6, I_7-9, M_4-6, M_

Record: 2 di 4177

Figure 3.2 The result of the SOs exportation to database when the matching comparison is stored in the 'SO' attribute as either (a) one record for each single matching or (b) one record for multiple matching.

or one record for multiple matching (see Figure 3.2). In the former case, for each SO s matching an individual $I \in \Omega$, a row describing I is inserted in the output database table and the 'SO' attribute contains the identifier (name or label) of s as value. This means that when several SOs match the same individual $I \in \Omega$, the output database table contains one row for each SO matching I. In the latter case, a single row of the output database table describes the non-empty list of SOs matching I. In this case, the value assigned to the 'SO' attribute is the list of the identifiers of the SOs matching I.

3.5 An application of SO2DB

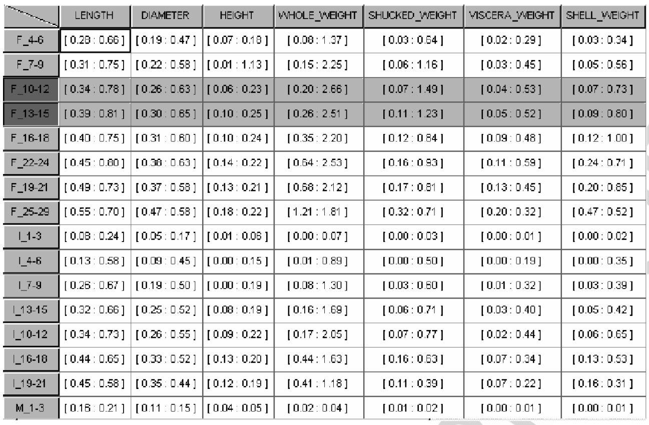
In this section, we show how SO2DB is used to export 24 SOs stored in the ASSO file abalone.xml to the abalone database.

The symbolic data involved in this study were generated with DB2SO by generalizing the micro-data collected in the abalone database.[1] In particular, the abalone database contains 4177 cases of marine crustaceans described in terms of nine attributes, namely, sex (discrete), length (continuous), diameter (continuous), height (continuous), whole weight (continuous), shucked weight (continuous), viscera weight (continuous), shell weight (continuous), and number of rings (integer). Symbolic data are then generated by the Cartesian product of sex (F=female, M=male, I=infant) and the range of values for the number of rings ([4, 6], [7, 9], [10, 12], [13, 15], [16, 18], [19, 21], [22, 24], [25, 29]) and resulting SOs are described according to seven interval variables derived from generalizing the continuous attributes length, diameter, height, whole weight, shucked weight, viscera weight, and shell weight.

The file abalone.xml is opened by selecting File/Open from the menu bar of the SO2DB graphical user interface. A wizard allows users to select a subset of the symbolic variables reported in the ASSO file and identify one or more SOs from the ASSO file to be exported to database.

The database is accessed using the Microsoft ODBC facility. In particular, the wizard allows users to choose or create an ODBC data source to access the database, select a table from the list of tables collected in the database and interactively establish a correct

[1] The abalone database is available at the UCI Machine Learning Repository (http://www.ics.uci.edu/~mlearn/MLRepository.html).



	LENGTH	DIAMETER	HEIGHT	WHOLE_WEIGHT	SHUCKED_WEIGHT	VISCERA_WEIGHT	SHELL_WEIGHT
F_4-6	[0.28 : 0.66]	[0.19 : 0.47]	[0.07 : 0.18]	[0.08 : 1.37]	[0.03 : 0.64]	[0.02 : 0.29]	[0.03 : 0.34]
F_7-9	[0.31 : 0.75]	[0.22 : 0.58]	[0.01 : 1.13]	[0.15 : 2.25]	[0.06 : 1.16]	[0.03 : 0.45]	[0.05 : 0.56]
F_10-12	[0.34 : 0.78]	[0.26 : 0.63]	[0.06 : 0.23]	[0.20 : 2.66]	[0.07 : 1.49]	[0.04 : 0.53]	[0.07 : 0.73]
F_13-15	[0.39 : 0.81]	[0.30 : 0.65]	[0.10 : 0.25]	[0.26 : 2.51]	[0.11 : 1.23]	[0.05 : 0.52]	[0.09 : 0.80]
F_16-18	[0.40 : 0.75]	[0.31 : 0.60]	[0.10 : 0.24]	[0.35 : 2.20]	[0.12 : 0.84]	[0.09 : 0.48]	[0.12 : 1.00]
F_22-24	[0.45 : 0.80]	[0.38 : 0.63]	[0.14 : 0.22]	[0.64 : 2.53]	[0.16 : 0.93]	[0.11 : 0.59]	[0.24 : 0.71]
F_19-21	[0.49 : 0.73]	[0.37 : 0.58]	[0.13 : 0.21]	[0.68 : 2.12]	[0.17 : 0.81]	[0.13 : 0.45]	[0.20 : 0.85]
F_25-29	[0.55 : 0.70]	[0.47 : 0.58]	[0.18 : 0.22]	[1.21 : 1.81]	[0.32 : 0.71]	[0.20 : 0.32]	[0.47 : 0.52]
I_1-3	[0.08 : 0.24]	[0.05 : 0.17]	[0.01 : 0.06]	[0.00 : 0.07]	[0.00 : 0.03]	[0.00 : 0.01]	[0.00 : 0.02]
I_4-6	[0.13 : 0.58]	[0.09 : 0.45]	[0.00 : 0.15]	[0.01 : 0.89]	[0.00 : 0.50]	[0.00 : 0.19]	[0.00 : 0.35]
I_7-9	[0.26 : 0.67]	[0.19 : 0.50]	[0.00 : 0.19]	[0.08 : 1.30]	[0.03 : 0.60]	[0.01 : 0.32]	[0.03 : 0.39]
I_13-15	[0.32 : 0.66]	[0.25 : 0.52]	[0.08 : 0.19]	[0.16 : 1.69]	[0.06 : 0.71]	[0.03 : 0.40]	[0.05 : 0.42]
I_10-12	[0.34 : 0.73]	[0.26 : 0.55]	[0.09 : 0.22]	[0.17 : 2.05]	[0.07 : 0.77]	[0.02 : 0.44]	[0.06 : 0.65]
I_16-18	[0.44 : 0.65]	[0.33 : 0.52]	[0.13 : 0.20]	[0.44 : 1.63]	[0.16 : 0.63]	[0.07 : 0.34]	[0.13 : 0.53]
I_19-21	[0.45 : 0.58]	[0.35 : 0.44]	[0.12 : 0.19]	[0.41 : 1.18]	[0.11 : 0.39]	[0.07 : 0.22]	[0.16 : 0.31]
M_1-3	[0.16 : 0.21]	[0.11 : 0.15]	[0.04 : 0.05]	[0.02 : 0.04]	[0.01 : 0.02]	[0.00 : 0.01]	[0.00 : 0.01]

Figure 3.3 Symbolic descriptions of the SOs stored in abalone.xml.

association between the symbolic variables reported in the input ASSO file and the columns of this database table.

Finally, users specify the type of matching (canonical or flexible), the name of the output database table describing the matching individuals and the format of the matching result (i.e., either one record for each single matching or one record for multiple matching).

In this study, we decided to export the 24 SOs underlying the symbolic descriptions stored in abalone.xml to the same micro-data processed to generate the symbolic data. All the symbolic variables associated to the columns of the symbolic data table are selected and the canonical matching is chosen to determine the abalones that exactly match each SO.

Finally, the extents resulting from exporting abalone SOs to the abalone database are compared to identify SOs covered by the same individuals. For instance, we may compare the extents computed on the abalone database of the SO ‘F_10-12’, whose description is generated by generalizing the abalones with ‘sex = F’ and ‘number of rings ∈ [10, 12]’, and the SO ‘F_13-15’, whose description is generated by generalizing the abalones with ‘sex = F’ and ‘number of rings ∈ [13, 15]’ (see Figure 3.3). The export of ‘F_10-12’ to the abalone database retrieves 3650 cases, while the export of ‘F_13-15’ to the abalone database retrieves 3391 cases. When we compare the extents of ‘F_10-12’ and ‘F_13-15’, we discover that they share exactly 3379. This may suggest a higher level of aggregation (i.e., ‘F_10-15’) in generating the SOs.

References

Bock, H.-H. and Diday, E. (2000) Symbolic objects. In H.-H. Bock and E. Diday (eds), *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data*, pp. 54–77. Berlin: Springer-Verlag.

Diday, E. and Esposito, F. (2003) An introduction to symbolic data analysis and the SODAS software. *Intelligent Data Analysis* 7(6), 583–602.

Esposito, F., Malerba, D. and Lisi, F.A. (2000) Matching symbolic objects. In H.-H. Bock and E. Diday (eds), *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data*, pp. 186–197. Berlin: Springer-Verlag.